Fast Fourier Transform (FFT)

Problem: we need an efficient way to compute the DFT. The answer is the FFT.

Consider a data sequence $x = [x(0), x(1), \ldots, x(N-1)]$ *and its DFT:*

$$X(k) = \sum_{n=0}^{N-1} x(n) {w_N}^{kn}, \quad k = 0, \ldots, N-1$$

We can always break the summation into two summations: one on even indices (n=0,2,4,…) and one on odd indeces (n=1,3,5,…), as

$$X(k) = \sum_{n\,even} x(n) {w_N}^{kn} + \sum_{n\,odd} x(n) {w_N}^{kn}, \quad k = 0, \ldots, N-1$$

Let us assume that the total number of points N is *<u>even</u>*, *ie N/2 is an integer. Then we can write the DFT as*

$$X(k) = \sum_{n\,even} x(n) w_N^{\;kn} + \sum_{n\,odd} x(n) w_N^{\;kn}$$

$$= \sum_{m=0}^{\frac{N}{2}-1} x(2m) w_N^{\;k(2m)} + \sum_{m=0}^{\frac{N}{2}-1} x(2m+1) w_N^{\;k(2m+1)}$$

$$= \sum_{m=0}^{\frac{N}{2}-1} x(2m) \underbrace{\left(w_N^{\;2}\right)}_{w_{N/2}}{}^{km} + w_N^{\;k} \sum_{m=0}^{\frac{N}{2}-1} x(2m+1) \underbrace{\left(w_N^{\;2}\right)}_{w_{N/2}}{}^{km}$$

since $\quad w_N^{\;2} = \left(e^{-j2p/N}\right)^2 = e^{-j2p/(N/2)} = w_{N/2}$

The two summations are <u>two distinct DFT's</u>, as we can see below

$$X(k) = \sum_{m=0}^{\frac{N}{2}-1} x(2m) w_{N/2}^{\,km} + w_N^{\,k} \sum_{m=0}^{\frac{N}{2}-1} x(2m+1) w_{N/2}^{\,km}$$

$$\boxed{\text{N-point DFT}} = \boxed{\text{N/2-point DFT}} + w_N^{\,k} \boxed{\text{N/2-point DFT}}$$

$$X_N(k) = X^e{}_{N/2}(k) + w_N^{\,k} X^o{}_{N/2}(k)$$

for k=0,…N-1, where

$$X^e{}_{N/2} = DFT\left[x^{even}\right], \; x^{even} = \left[x(0), x(2), \ldots, x(N-2)\right];$$

$$X^o{}_{N/2} = DFT\left[x^{odd}\right], \; x^{odd} = \left[x(1), x(3), \ldots, x(N-1)\right].$$

The problem is that in the expression

$$X_N(k) \quad = \quad X^e{}_{N/2}(k) \quad + \quad w_N{}^k X^o{}_{N/2}(k)$$

the N-point DFT and the N/2 point DFT's have different lengths, since we define them as

$$X_N(k), \quad k = 0,\ldots, N-1$$

$$X_{N/2}(k), \quad k = 0,\ldots, N/2-1$$

For example if N=4 we need to compute

$$X_4 = [X_4(0),\ X_4(1),\ X_4(2),\ X_4(3)]$$

from two 2-point DFT's

$$X^e = [X_2{}^e(0),\ X_2{}^e(1)]$$

$$X^o = [X_2{}^o(0),\ X_2{}^o(1)]$$

So how do we compute $X_4(2)$ *and* $X_4(3)$ *?*

We use the periodicity of the DFT, and relate the N-point DFT with the two N/2-point DFT's as follows

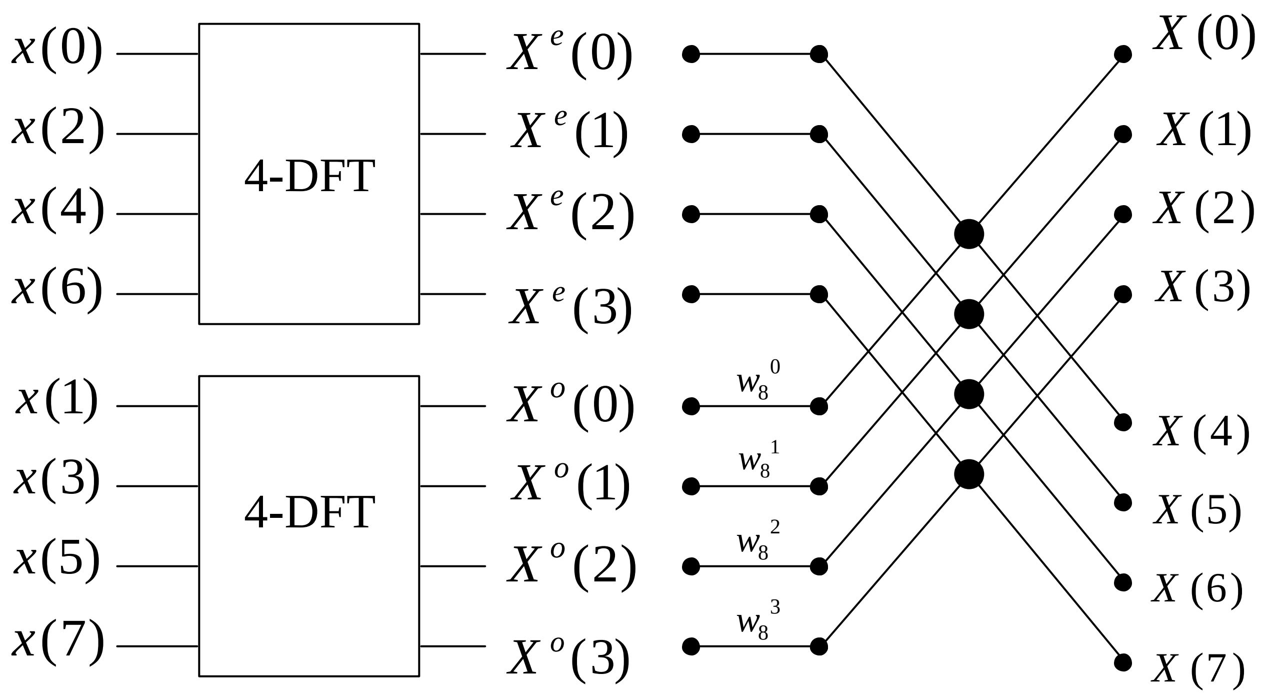
$$\boxed{\begin{aligned} X_N(k) &= X^e_{N/2}(k) + w_N{}^k X^o_{N/2}(k) \\ X_N\left(k+\frac{N}{2}\right) &= X^e_{N/2}(k) - w_N{}^k X^o_{N/2}(k), \qquad k = 0,\ldots,\frac{N}{2}-1 \end{aligned}}$$

where we used the facts that

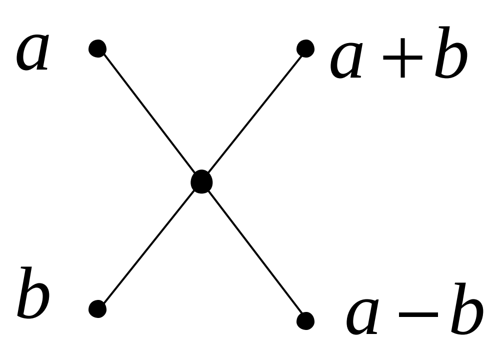
- *the DFT is periodic and in particular* $X_{N/2}(k) = X_{N/2}\left(k+\frac{N}{2}\right);$

- $w_N{}^{k+N/2} = w_N{}^k e^{-(j2p/N)N/2} = -w_N{}^k$

General Structure of the FFT (take, say, N=8):

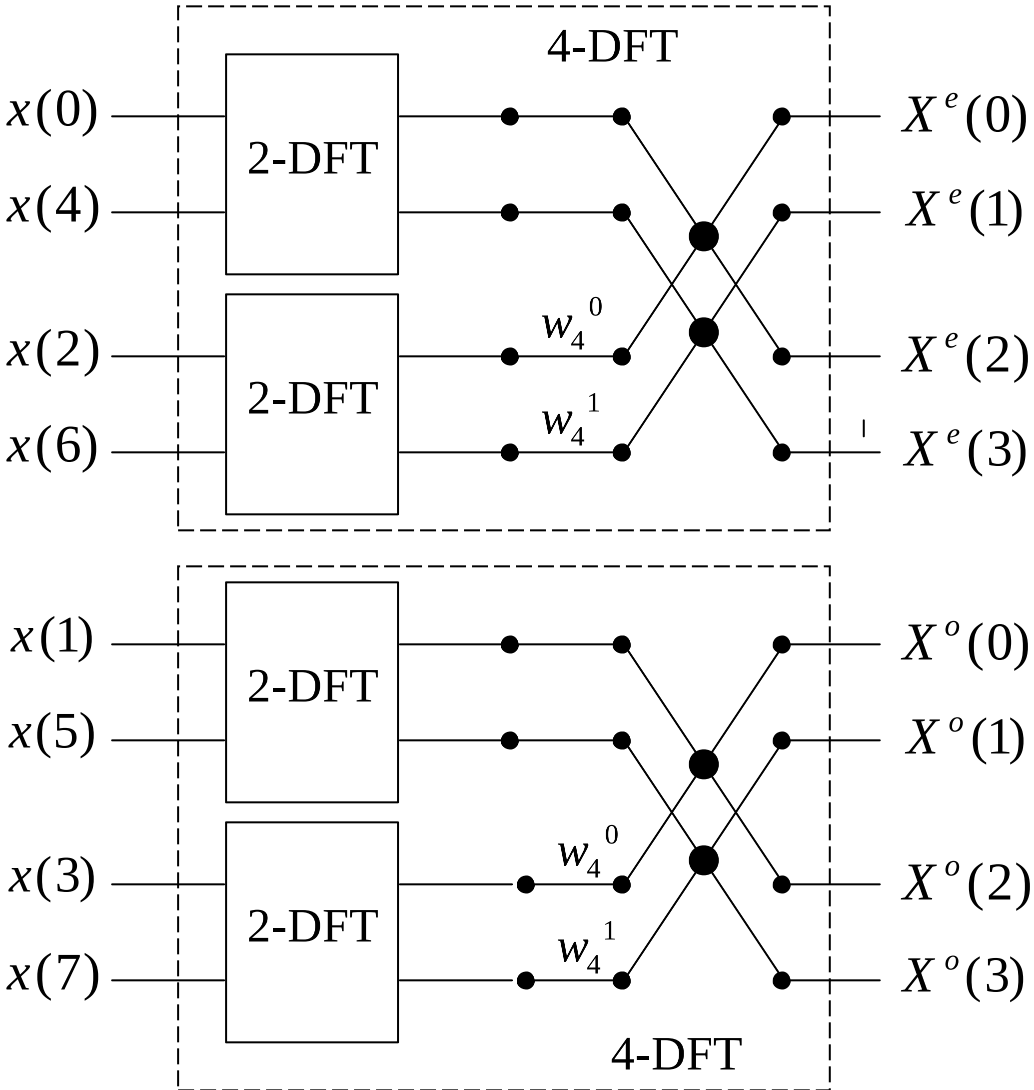


where

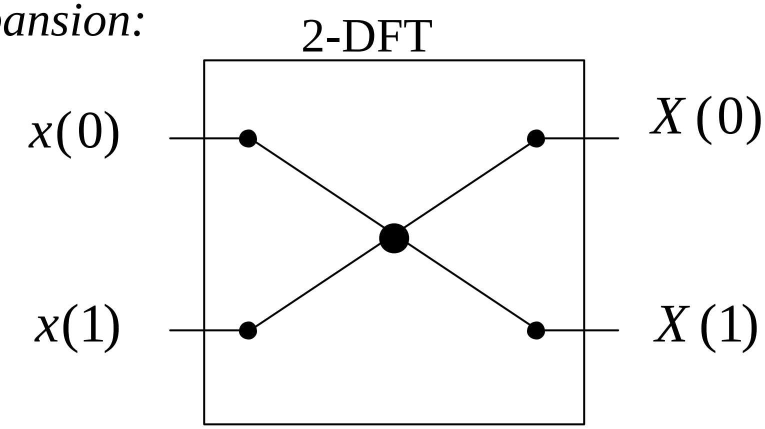


is called the <u>*butterfly*</u>.

Same for the 4-DFT:



Finally the 2-DFT's have a simple expansion:



$$X_2(0) = x(0) + x(1)$$

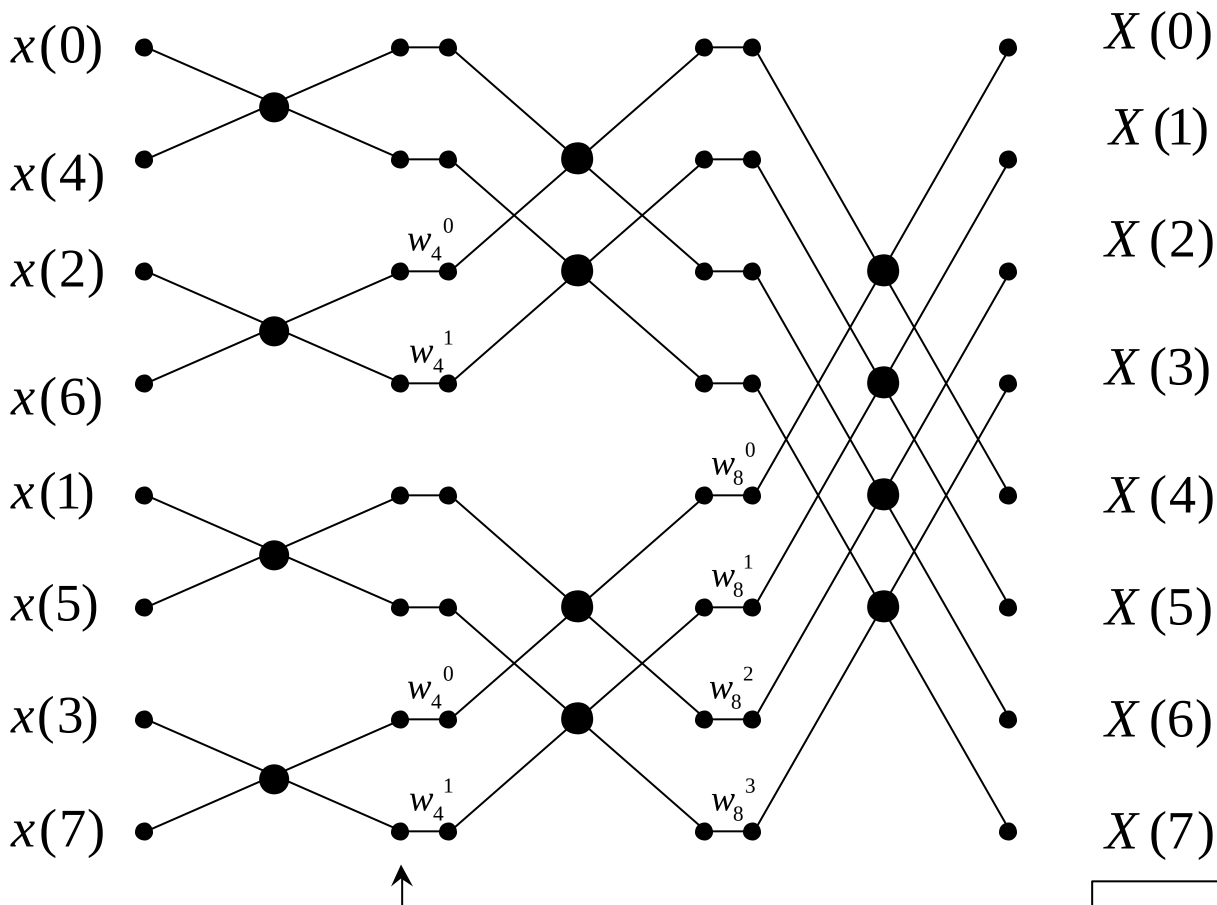
$$X_2(1) = x(0) - x(1),$$

since

$$X_2(k) = \sum_{n=0}^{1} x(n) w_2^{kn} = x(0) + w_2^{k} x(1)$$

with $$w_2^{k} = e^{-j\pi k} = (-1)^k$$

Put everything together:



N/2 mult/stage

N adds/stage

$\times$ $\log_2 N \mathit{stages}$ $=$

$\frac{N}{2}\log_2 N$ mult

$N\log_2 N$ adds

We say that, for a data set of length $N = 2^L$,

$$\boxed{\textit{complexity of the FFT is}\ O\left\{ N \log_2 N \right\}}$$

i.e. $\quad$ $\textit{number of operations} \le a\, N \log_2 N + b$ $\quad$ *for some constants a,b.*

On the other hand, for the same data of length N ,

$$\boxed{\textit{complexity of the DFT is}\ O\left\{ N^2 \right\}}$$

Since, from the formula, $\quad X(k) = \sum_{n=0}^{N-1} x(n) w_N{}^{kn}, \quad k = 0, \ldots, N-1$

$$\boxed{\text{N ops/term}} \times \boxed{\text{N terms}} = \boxed{\text{N}^2 \text{ ops}}$$

This is a big difference in the total number of computations, as shown in this graph: